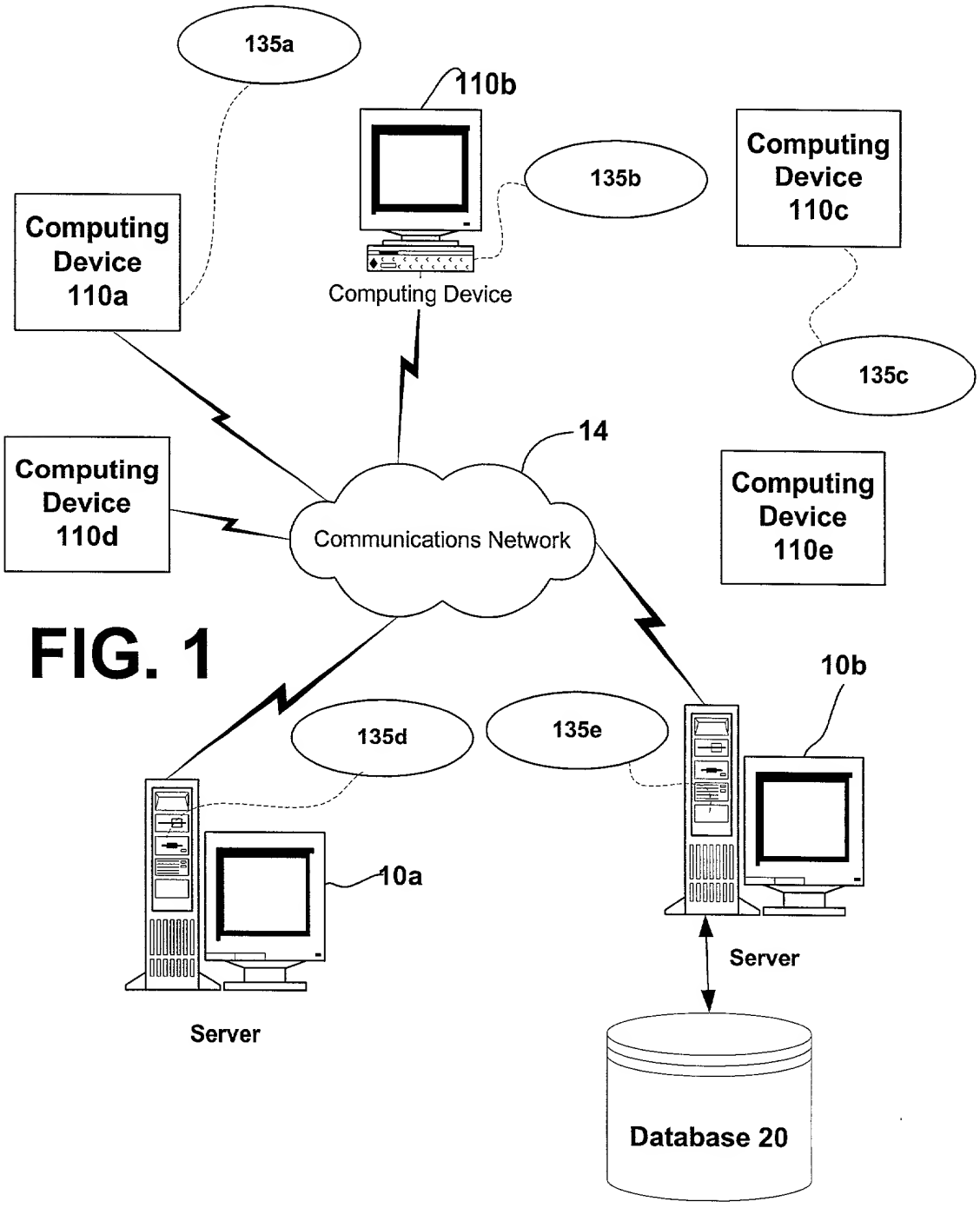
135a
110b
135b
Computing Device 110c
Computing Device 110a
Computing Device
135c
14
Computing Device 110d
Communications Network
Computing Device 110e
10b
135d
135e
10a
Server
Server
Database 20

FIG. 1

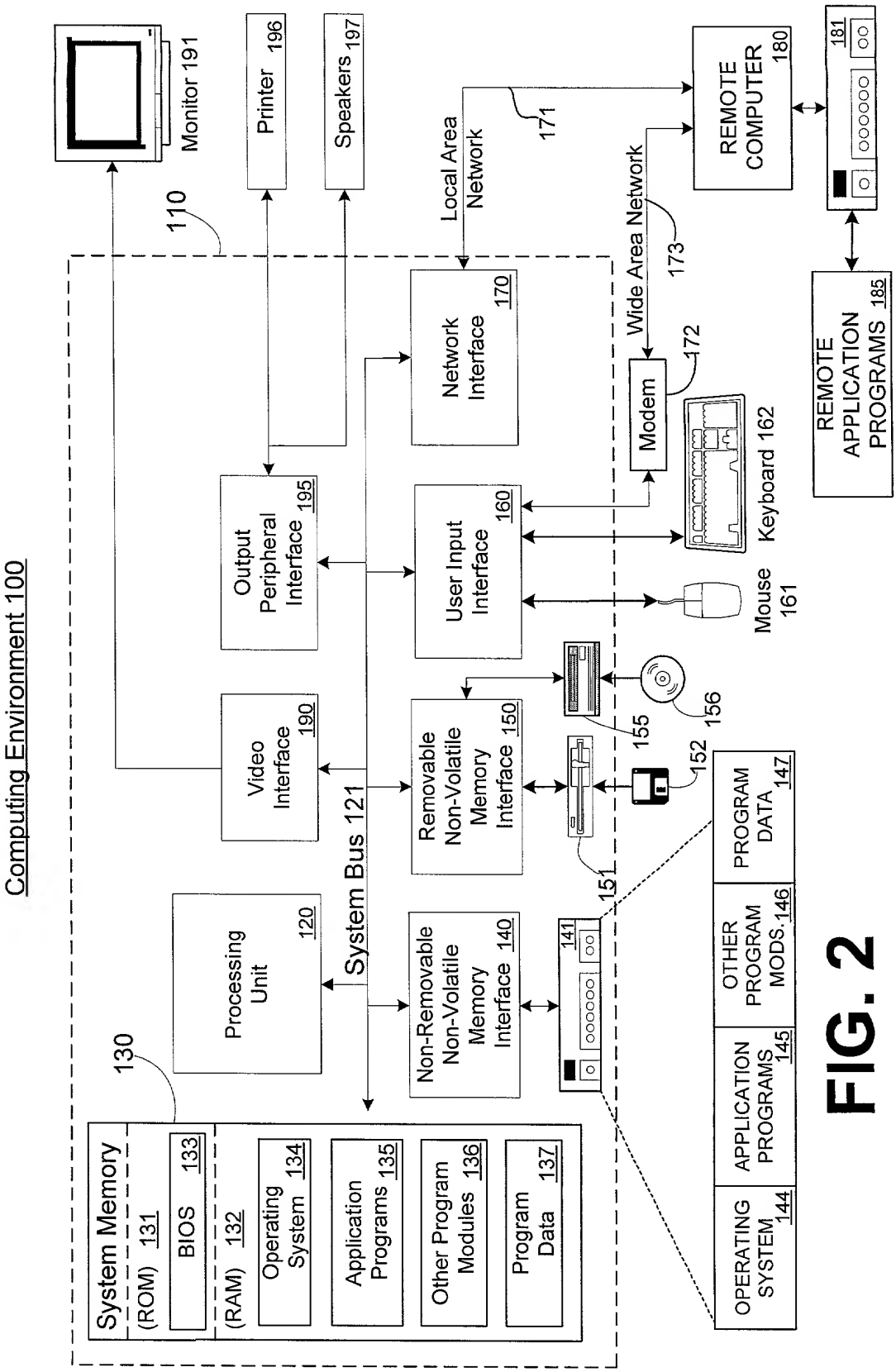
Computing Environment 100
110
Monitor 191
Printer 196
Speakers 197
Local Area Network
171
REMOTE COMPUTER 180
181
Wide Area Network
173
Network Interface 170
Modem 172
REMOTE APPLICATION PROGRAMS 185
Output Peripheral Interface 195
User Input Interface 160
Keyboard 162
Mouse 161
Video Interface 190
Removable Non-Volatile Memory Interface 150
155
156
152
151
System Bus 121
Processing Unit 120
Non-Removable Non-Volatile Memory Interface 140
141
130
System Memory
(ROM) 131
BIOS 133
(RAM) 132
Operating System 134
Application Programs 135
Other Program Modules 136
Program Data 137
OPERATING SYSTEM 144
APPLICATION PROGRAMS 145
OTHER PROGRAM MODS. 146
PROGRAM DATA 147

FIG. 2

FIG. 3A

300

```
using System;
namespace FirstParty.Component {
        public class A {
                public virtual void F() {
                        Console.WriteLine("A.F()");
                }
        }
}
```

FIG. 3B

302

```
using System;
using FirstParty.Component;
namespace SecondParty.Application {
        class B: A {
                public void F(long count) {
                        for (int i = 0; i < count; i++)
                                Console.WriteLine("B.F(int)");
                }
                public void G() {
                        Console.WriteLine("B.G()");
                }
                static void Main() {
                        B b = new B();
                        b.F();
                        b.F(1);
                        b.G();
                }
        }
}
```

400

```
using System;
namespace FirstParty.Component {
        public class A {
        }
}
```

402a

FIG. 4A

410

```
using System;
using FirstParty.Component;
namespace SecondParty.Application {
        class B: A {
                public void G() {
                        Console.WriteLine("B.G");
                }
                static void Main() {
                        B b = new B();
                        b.G();
                }
        }
}
```

412

414a

FIG. 4B

420

```
// version 2
using System;
namespace FirstParty.Component {
        public class A {
                public virtual void G() {
                        Console.WriteLine("A.G");
                }
        }
}
```

402b

414b

FIG. 4C

430

```
// version 2
using System;
using FirstParty.Component;
namespace SecondParty.Application {
        class B: A {                                    414b
                new public void G() {
        432             Console.WriteLine("B.G");
                }
                static void Main() {
                        B b = new B();
                        b.G();
                }
        }
}
```

FIG. 4D

440

```
using System;
using FirstParty.Component;
namespace SecondParty.Application {
        class B: A {                                    414b
                override public void G() {
        442             Console.WriteLine("B.G");
                }
                static void Main() {
                        B b = new B();
                        b.G();
                }
        }
}
```

FIG. 4E

500

```
using System;
class Test {
        static void F() {
                Console.WriteLine("Test.F");
        }
        static void F(int i) {
                Console.WriteLine("Test.F(int)");
        }
        static void Main() {
                F();
                F(3);
        }
}
```

FIG. 5A

FIG. 5B

```
Test.F
Test.F(int)
```

502

510

```
using System;
class A {
        public void F() {
                Console.WriteLine("Test.F");
        }
}
class B: A {
        public void F(int i) {
                Console.WriteLine("Test.F(int)");
        }
}
class Test {
        static void Main() {
                int i = 3;
                B b = new B();
                b.F();
                b.F(i);
        }
}
```

FIG. 5C

FIG. 5D

520

```
using System;
namespace FirstParty.Component {
        public class A {
        }
}
```

FIG. 5E

530

```
using System;
using FirstParty.Component;
namespace SecondParty.Application {
        class B: A {
                public void F(long count) {
                        for (int i = 0; i < count; i++)
                                Console.WriteLine("B.F(int)");
                }
                static void Main() {
                        int i = 3;
                        B b = new B();
                        b.F(i);
                }
        }
}
```

FIG. 5F

```
B.F(int)
```

540

600

```
class Base
{
        public void F() {}
}
class Derived: Base
{
        public void F() {}              // Warning,
hiding an inherited name
}
```

FIG. 6A

610

```
class Base
{
        public void F() {}
}
class Derived: Base
{
        new public void F() {}
}
```

FIG. 6B

FIG. 6C

620

```
class Base
{
        public static void F() {}
}
class Derived: Base
{
        new private static void F() {}        // Hides Base.F in Derived only
}
class MoreDerived: Derived
{
        static void G() { F(); }                    // Invokes Base.F
}
```

FIG. 7

700

```
interface ITest
{
        void F();                          // F()
        void F(int x);                     // F(int)
        void F(ref int x);                 // F(ref int)
        void F(out int x);                 // F(out int)
        void F(int x, int y);              // F(int, int)
        int F(string s);                   // F(string)
        int F(int x);                      // F(int)
        void F(string[] a);                // F(string[])
        void F(params string[] a);         // F(string[])
}
```

800

```
class A
{
        public void F() {
Console.WriteLine("A.F"); }
        public virtual void G() {
Console.WriteLine("A.G"); }
}
class B: A
{
        new public void F() {
Console.WriteLine("B.F"); }
        public override void G() {
Console.WriteLine("B.G"); }
}
class Test
{
        static void Main() {
                B b = new B();
                A a = b;
                a.F();
                b.F();
                a.G();
                b.G();
        }
}
```

FIG. 8A

810

```
A.F
B.F
B.G
B.G
```

FIG. 8B

```
class A
{
        public virtual void F() {
Console.WriteLine("A.F"); }
}
class B: A
{
        public override void F() {
Console.WriteLine("B.F"); }
}
class C: B
{
        new public virtual void F() {
Console.WriteLine("C.F"); }
}
class D: C
{
        public override void F() {
Console.WriteLine("D.F"); }
}
class Test
{
        static void Main() {
                D d = new D();
                A a = d;
                B b = d;
                C c = d;
                a.F();
                b.F();
                c.F();
                d.F();
        }
}
```

820

FIG. 8C

830

```
B.F
B.F
D.F
D.F
```

FIG. 8D

```
class A
{
        int x;
        public virtual void PrintFields() {
                Console.WriteLine("x = {0}", x);
        }
}
class B: A
{
        int y;
        public override void PrintFields() {
                base.PrintFields();
                Console.WriteLine("y = {0}", y);
        }
}
```

900

FIG. 9A

```
class A
{
        public virtual void F() {}
}
class B: A
{
        public virtual void F() {}              //
Warning, hiding inherited F()
}
```

910

FIG. 9B

```
class A
{
        public virtual void F() {}
}
class B: A
{
        new private void F() {}        // Hides A.F within B
}
class C: B
{
        public override void F() {}    // Ok, overrides A.F
}
```

920

FIG. 9C